B 群

配列のペアワイズなローカルアライメント．対象は DNA でもアミノ酸配列でも可．

次の仕様で Local alignment algorithm を実装した．

- MATCH_AWARD = 2，DISMATCH_PENALTY = -1，GAP_PENALTY = -1，はそれぞれ match，unlatch，gap の採点
- m,n はそれぞれ文字列 sa，sb の長さ，sa[i]，sb[j] はそれぞれ sa，sb の要素
- score は採点行列 H，pointer は最適方向行列

1 アルゴリズム

1. 初期化：
   $H(i,0) = 0, 0 \leq i \leq m;\quad H(0,j) = 0, 0 \leq j \leq n$
   max score = 0
2. 採点行列 H と最適方向行列 D を計算（各 $(i,j), 1 \leq i \leq m, 1 \leq j \leq n$）：
   d=match/unmatch の時 $H(i-1,j-1)$ を加点・ペナルティした点数；
   l=delection の時の $H(i-1,j)$ + GAP_PENALTY；
   u=insertion の時の $H(i,j-1)$ + GAP_PENALTY
   $H(i,j)$ =MAX(d,l,u,0)
   $D(i,j)$ = 高い点数の方向，優先順位は対角，上（insertion），左（delection），点数が 0 ならばストップ
   max score を更新する，max score の位置を記録する．
3. trace back
   max score の位置から最適方向行列 D で記録した方向を用いて文字列を生成する，

2 テスト例

テスト用文字列はそれぞれ wikipedia と配布資料#3 Sequene Alignment の例である．

- 入力：文字列 sa,sb
- 出力：採点行列 H，最適アライメントの座標表示，最高点数，最適アライメントの文字列

1. # input
   sa="ACACACTA"
   sb="AGCACACA"
   # output
   [[0. 0. 0. 0. 0. 0. 0. 0. 0.]
    [0. 2. 1. 2. 1. 2. 1. 0. 2.]
    [0. 1. 1. 1. 1. 1. 1. 0. 1.]
    [0. 0. 3. 2. 3. 2. 3. 2. 1.]
    [0. 2. 2. 5. 4. 5. 4. 3. 4.]
    [0. 1. 4. 4. 7. 6. 7. 6. 5.]
    [0. 2. 3. 6. 6. 9. 8. 7. 8.]
    [0. 1. 4. 5. 8. 8. 11. 10. 9.]
    [0. 2. 3. 6. 7. 10. 10. 10. 12.]]
   cell(8,8):diagonal
   cell(7,7):left
   cell(6,7):diagonal
   cell(5,6):diagonal
   cell(4,5):diagonal
   cell(3,4):diagonal
   cell(2,3):diagonal
   cell(1,2):up
   cell(1,1):diagonal
   max score:12
   result:
   A-CACACTA
   AGCACAC-A
2. # input
   sa="GCTCGTTG"
   sb="AACCGTAA"
   # output
   [[0. 0. 0. 0. 0. 0. 0. 0. 0.]
    [0. 0. 0. 0. 0. 0. 0. 0. 0.]
    [0. 0. 0. 0. 0. 0. 0. 0. 0.]
    [0. 0. 2. 1. 2. 1. 0. 0. 0.]
    [0. 0. 2. 1. 3. 2. 1. 0. 0.]
    [0. 2. 1. 1. 2. 5. 4. 3. 2.]
    [0. 1. 1. 3. 2. 4. 7. 6. 5.]
    [0. 0. 0. 2. 2. 3. 6. 6. 5.]

```
 [ 0.  0.  0.  1.  1.  2.  5.  5.  5.]]
cell(6,6):diagonal
cell(5,5):diagonal
cell(4,4):diagonal
cell(3,3):left
cell(2,3):diagonal
max score:7
result:
CTCGT
C-CGT
```